WEST Search History

DATE: Friday, November 21, 2003

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT,PGPB; PLUR=YES; OP=ADJ		
L23	L20 and (switch$ or swrap$)	4	L23
L22	L21 and (switch$ or swrap$)	63	L22
L21	L19 and l13	109	L21
L20	L19 and l6	17	L20
L19	l18 or l17 or l16	4201	L19
L18	((712/228 |712/229 |712/232 |712/233 |712/234 |712/235 |712/236 |712/237)!.CCLS.)	1421	L18
L17	((709/106 |709/107 |709/108)!.CCLS.)	977	L17
L16	((717/106 |717/136 |717/137 |717/138 |717/139 |717/140 |717/141 |717/142 |717/143 |717/144 |717/145 |717/146 |717/147 |717/148 |717/149 |717/151 |717/152 |717/153 |717/159 |717/160 |717/161)!.CCLS.)	1948	L16
	DB=USPT; PLUR=YES; OP=ADJ		
L15	built-in near compil$ and (processor or coprocessor)	10	L15
L14	L13 and legacy near (software or program or code)	7	L14
L13	(optimiz$ near2 (co?processor or coprocessor or processor))	1295	L13
	DB=EPAB,DWPI,TDBD; PLUR=YES; OP=ADJ		
L12	L11	182	L12
	DB=JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=ADJ		
L11	(optimiz$ near2 (co?processor or coprocessor or processor))	243	L11
L10	(optimiz$ near2 code)and (co?processor or coprocessor)	2	L10
L9	(optimiz$ near2 code)and (co?processor or processor)	98	L9
L8	L7 and (optimiz$ near2 code)	1	L8
L7	legacy near (software or program or code)	74	L7
	DB=USPT,PGPB; PLUR=YES; OP=ADJ		
L6	L4 and (optimiz$ near2 code)	63	L6
L5	L4 and (optimiz$ near2 code) and coprocessor	1	L5
L4	legacy near (software or program or code)	526	L4
	DB=USPT; PLUR=YES; OP=ADJ		
L3	L2	79	L3
	DB=USPT,PGPB; PLUR=YES; OP=ADJ		
L2	L1 and cache	98	L2
L1	(optimiz$ near2 code) and coprocessor	121	L1

END OF SEARCH HISTORY

Search Results - Record(s) 1 through 4 of 4 returned.

☐ 1. Document ID: US 20030110478 A1

L23: Entry 1 of 4 File: PGPB Jun 12, 2003

PGPUB-DOCUMENT-NUMBER: 20030110478
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20030110478 A1

TITLE: Portable run-time code synthesis in a caching dynamic translator

PUBLICATION-DATE: June 12, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Duesterwald, Evelyn	Somerville	MA	US	
Desoli, Giuseppe	Watertown	MA	US	
Bala, Vasanth	Sudbury	MA	US	

US-CL-CURRENT: 717/137

ABSTRACT:

A method of producing a caching dynamic translator with portable run-time code synthesis includes programming hardware independent replacement functions in a high level programming language for the caching dynamic translator, and compiling the hardware independent replacement functions to produce hardware dependent computer executable replacement functions.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Image

☐ 2. Document ID: US 20020147970 A1

L23: Entry 2 of 4 File: PGPB Oct 10, 2002

PGPUB-DOCUMENT-NUMBER: 20020147970
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20020147970 A1

TITLE: Method and system for optimizing code using an optimizing coprocessor

PUBLICATION-DATE: October 10, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Smith, Jack Robert	South Burlington	VT	US	
Ventrone, Sebastian Theodore	South Burlington	VT	US	

US-CL-CURRENT: 717/140

ABSTRACT:

A data processing system includes a central processing unit (CPU) in communication with a system memory. Within the system memory, there is stored legacy code that does not utilize the full features of the CPU. The data processing system also includes a code-optimizing coprocessor in communication with the CPU and the system memory. Control logic within the code-optimizing coprocessor causes the code-optimizing coprocessor to generate optimized code from the legacy code at the same time the CPU executes the legacy code, such that the optimized code is tailored according to the CPU. After the code-optimizing coprocessor has generated at least some optimized code, the code-optimizing coprocessor causes the CPU to automatically utilize at least some optimized code in lieu of at least some of the legacy code.

☐ 3. Document ID: US 6631514 B1

L23: Entry 3 of 4 File: USPT Oct 7, 2003

US-PAT-NO: 6631514
DOCUMENT-IDENTIFIER: US 6631514 B1

TITLE: Emulation system that uses dynamic binary translation and permits the safe speculation of trapping operations

DATE-ISSUED: October 7, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Le; Bich-Cau	San Jose	CA		

US-CL-CURRENT: 717/137; 717/136, 717/151, 717/159

ABSTRACT:

The inventive emulator dynamically translates instructions in code written for a first architecture into code for a second architecture. The emulator designates various checkpoints in the original code, and speculatively reorders the placement of the translated code instructions according to optimization procedures. If during the execution of the reordered code, a trap should occur, then the emulator resets the original code to the most recent checkpoint and begins executing the original code sequentially in a line-by-line manner until the section is completed or branched out of. The original code is reset by changing the program counter to the checkpoint, and reversing the effects of each instruction which has been executed subsequent to the checkpoint. Thus, any native instructions which correspond to original instructions which occur sequentially prior to the checkpoint have been executed, and any native instructions which correspond to original instructions which occur sequentially subsequent to the checkpoint have not been executed.

52 Claims, 9 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 3

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

---

☐ 4. Document ID: US 6301705 B1

L23: Entry 4 of 4 File: USPT Oct 9, 2001

US-PAT-NO: 6301705
DOCUMENT-IDENTIFIER: US 6301705 B1

TITLE: System and method for deferring exceptions generated during speculative execution

DATE-ISSUED: October 9, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Doshi; Gautam B.	Sunnyvale	CA		
Markstein; Peter	Woodside	CA		
Karp; Alan H.	Palo Alto	CA		
Huck; Jerome C.	Palo Alto	CA		
Colon-Bonet; Glenn T.	Fort Collins	CA		
Morrison; Michael	Sunnyvale	CA		

US-CL-CURRENT: 717/154; 712/208, 712/222, 712/244, 717/161

ABSTRACT:

The present invention is generally directed to a system and method for supporting speculative execution of an instruction set for a central processing unit (CPU) including non-speculative and speculative instructions. In accordance with one aspect of the invention a method includes the steps of evaluating the instructions of the program to determine whether the individual instructions are speculative or non-speculative, and assessing each of the speculative instructions to determine whether it generates an exception. For each of the speculative instructions that generates an exception, the method then encode a deferred exception token (DET) into an unused register value of a register of the CPU. In accordance with another aspect of the invention, a system is provided, which system includes circuitry configured to evaluate the instructions of the instruction set to determine whether the individual instructions are speculative or non-speculative. The system further includes circuitry configured to assess each of the speculative instructions to determine whether it generates an exception. Finally, the system further includes circuitry configured to encode a deferred exception token (DET) into an unused register value of a register of the (CPU.

17 Claims, 7 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4

Terms	Documents
L20 and (switch$ or swrap$)	4